PB-217 602

USER MANUAL FOR MAIDS PROGRAM: PROGRAM FOR DETERMINATION OF BETA-PREDICTION EQUATIONS BY MANIPULATION AND INTERPRETATION OF DATA SYSTEMS

Harold R. Bridges, et al

Iowa State University
Ames, Iowa

15 February 1968

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

WP-00196

PB 217 602

ENGINEERING
RESEARCH
INSTITUTE

IOWA STATE
UNIVERSITY
AMES, IOWA

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
U S Department of Commerce
Springfield VA 22151

58

ENGINEERING RESEARCH

ENGINEERING RESEARCH

ENGINEERING RESEARCH

ENGINEERING RESEARCH

ENGINEERING RESEARCH

USER MANUAL
FOR
MAIDS PROGRAM
Program for Determination of Beta–Prediction Equations by MAnipulation and Interpretation of Data Systems

Harold R. Bridges
FWPCA Research Fellow

E. Robert Baumann
Professor of Civil Engineering

FEBRUARY 15, 1968

ENGINEERING RESEARCH INSTITUTE
Sanitary Engineering Section of the
Department of Civil Engineering

Iowa State University
Ames

Project 684-S
ERI-116

Ia

CONTENTS

	Page
NOMENCLATURE	i
PROLOGUE	iii
INTRODUCTION	1
INPUT CARD FORMS	4
KAN Array Card	4
Comment and Label Cards	5
Data Cards	5
Transformation Card	6
V* Card	11
PRINT Card	11
REGRESSION Card	12
PACOCO Card	12
***** Card and * DATA Card	12
END Card	13
STOP Card	13
FORTRAN LISTING	14
MAIN Program	15
Subroutine FRAN	17
Subroutine DORIS	18
Subroutine MARIE	19
Subroutine ALICE	20
Subroutine ANN	22
Subroutine GWEN	23
Subroutine KATHY	26
Subroutine NORMA	27
Subroutine CORA	28
Subroutine RENE	29

I b

	Page
Subroutine DONNA	31
Subroutine TERI	32
EXAMPLE INPUT AND OUTPUT	34
REFERENCES	51
ACKNOWLEDGEMENT	52

I c

NOMENCLATURE*

	Equation	
Any septum	$H_p = q\nu\xi w/g$	(1)
Cylindrical septum	$H_c = (R_S\sigma/\phi)\ln(1 + R_S\ \phi\ X/R_o^2)$	(2)
	$L = \sqrt{R_o^2 + R_S\ \phi\ X} - R_S$	(3)
Flat septum[a]	$H_c = \sigma X$	(4)
	$L = L_p + \frac{\phi X}{2}$	(5)

where,

$\delta = Q/V_f$

$\sigma = q^2\nu\beta C_D/g$

$R_o = R_S + L_p$

$\phi = 2q\gamma_w C_D(10)^{-6}/\gamma_p$

$L_p = w/\gamma_p$

$X = t - (1 - e^{-\delta t})/\delta$

Symbol	Meaning	Dimension
A_s	Septum area	$[L^2]$
β	Filter cake resistance index or β-index	$[L^{-2}]$
C_D	Body feed concentration, ppm by weight	[–][b]
C_S	Suspended solids concentration, ppm by weight	[–][b]
δ[c]	Dilution rate, theoretically Q/V_f	$[T^{-1}]$

[a] Septum that does not exhibit increasing area effect

[b] Dimensionless

[c] δ will be equal to Q/V_f in a completely-mixed system in a filter housing. Since the conditions in practice are far from ideal, values of δ should be estimated by visual selection of the inflection points on the head loss versus time curve in any particular filter run with the assumption that at this time δt is approximately equal to 3.

Symbol	Meaning	Dimension
g	Gravity constant	$[LT^{-2}]$
γ_p	Bulk density of precoat	$[FL^{-3}]$
γ_c	Bulk density of filter cake	$[FL^{-3}]$
γ_w	Density of water	$[FL^{-3}]$
H_p	Headloss through precoat	$[L]$
H_c	Headloss through filter cake	$[L]$
L_p	Thickness of precoat	$[L]$
L_c	Thickness of filter cake	$[L]$
L	$L_p + L_c$	$[L]$
ν	Kinematic viscosity of influent	$[L^2T^{-1}]$
Q	Flow rate	$[L^3T^{-1}]$
q	Flow rate per unit septum area	$[LT^{-1}]$
R_S	Outer radius of septum	$[L]$
R_o	Outer radius of precoated septum = $R_S + L_p$	$[L]$
t	Time since start of filtering cycle of a particular run	$[T]$
V_f	Volume of housing	$[L^3]$
w	Precoat weight per unit area	$[FL^{-2}]$
X	$t - (1 - e^{-\delta t})/\delta$	$[T]$
ξ	Filter aid resistance index or ξ index	$[F^{-1}L]$

*Dillingham, J. H., Cleasby, J. L., and Baumann, E. R., "Diatomite Filtration Equations for Various Septa," Journal of the Sanitary Engineering Division, ASCE, Vol. 93, No. SA1, Proc. Paper 5115, February 1967, pp. 41-55.

PROLOGUE

Filter aid filters may be operated using one of several different materials such as diatomite or perlite as the filtering media. A long term study was undertaken at Iowa State University to determine the characteristics of municipal applications in which such filters may be used successfully, to study the effect of various variables on filtration economy and effectiveness, and to outline a procedure for designing a plant to operate with maximum economy. Filtration equations have been developed to predict the headloss-time relationship during a filter run by use of empirical filter cake resistance measurements. An equation has been developed to predict filter cake resistance from measureable filtration variables. Empirical constants in the prediction equation are collected in filter runs. Digital computer programs have been prepared to:

- Evaluate the filter cake resistance from the results of a single or series of filter runs (BID Program),
- Develop a filter cake resistance prediction equation from the results of several filter runs conducted with different levels of suspended solids and/or body feed (MAIDS Program), and
- Determine the proper design parameters of flow rate, terminal headloss, and filter aid body feed type and concentration which can be used to filter water at least cost (POPO Program).

This paper describes in brief the required MAIDS program listed in Fortran language, and describes its use with several examples.

INTRODUCTION

MAIDS (Manipulation and Interpretation of Data Systems) was designed to perform a multiple regression of from two to eight variables to fit the linear model,

$$T1 = B1 + B2*T2 + B3*T3 + \ldots + Bn*Tn$$

where T1 is the dependent variable, T2 ... Tn are independent variables, B1 .... Bn are regression coefficients. n can be from two to eight.

If the model (the equation of the curve being fitted) is not linear, some transformation or combination of transformations must be performed to make the model linear. For example, the model

$$T1 = 10^{B1} * T2^{B2} * T3^{B3}$$

can be made linear by taking the logarithm (base 10) of each variable.

$$\log T1 = B1 + B2*\log T2 + B3*\log T3.$$

Any desired transformation is possible with MAIDS. Thus, the program is very useful for reducing and printing out tabular data, even though a regression analysis is not desired.

ending page blank

80 COLUMN DATA SHEET

PROGRAM INPUT CARD FORMS | JOB NO. | BY | DATE

	1 (1–10)	2 (11–20)	3 (21–30)	4 (31–40)	5 (41–50)	6 (51–60)	7 (61–70)	8 (71–80)
1)	CLDTPAVRSEFOM	+-.0123456789	+-*/=$LSCEADMOJ	TXANIU				
2)	C L							
3)	X A	X a	X a	X a	X a	X a	X a	X a
4)	T	t	t	t	t	t	t	t
5)	V*	i						
6)	PRINT							
7)	REGRESSION							
8)	PACOCO							
9)	***** *DATA							
10)	END							
11)	STOP							

NOTE: X = Value of any observed variable (number)
a = Value of a single valued variable (number)
t = Transformation (from Table 1)
i = Any integer from 1 to 8

Fig. 1. Input card forms.

INPUT CARD FORMS

Data used by MAIDS is read into the computer on 80 column punched cards. The card formats used with MAIDS are shown in Fig. 1.

KAN Array Card

The KAN array card is always the first input card read by the computer. Each letter, symbol, and digit on the card is stored in an array named KAN. Therefore KAN(1) = C, KAN(2) = L, KAN(3) = D, ..., KAN(50) = U. The KAN array is then used by the computer to read the remaining input cards. This is done by comparing each letter, symbol, or digit on the input card with each element of the KAN array. For example, if the letter in the first column of an input card matches KAN(1), i.e. the letter C is punched in Column 1 of the input card, the card is a comment card, (see page 5). Therefore, the computer will print out the information contained on the card and then read the next input card.

Comment and Label Cards

Cards with C or L punched in Column 1 are ignored by MAIDS; however, the information on the cards is printed on the output sheet. These cards can be used to make comments or to label values in tables. Any characters can be punched in Columns 2 to 80 of a comment or label card and any number of comment cards can be used.

Data Cards

A card with Column 1 left blank is used to input observed values of the variables (T1, T2, T3, ..., Tn) used in the regression analysis. On an 80 column data card there are eight 10-column fields available for up to eight observed variables. The first field actually contains only nine columns since the first column must be left blank. Each observed variable may be punched anywhere within a particular 10-column field. The decimal point must occupy one column. A plus sign may or may not be punched before a positive valued variable. Each data card contains one observation of each variable being read in. A particular variable must be punched in the same 10-column field on all data cards. If a 10-column field on a particular data card is blank, that field is ignored; a blank is not interpreted as zero.

A card with A punched in Column 1 may be used to read from one to eight different constants into the computer. Each constant must be punched anywhere within one of the eight 10-column fields with the decimal point occupying one column. These constants are used in transforming the observed variables.

Transformation Card

A card with T punched in Column 1 is a transformation card. The letter t shown in Fig. 1 represents the transformation symbols given in the table on page 7.

The observed variables input on a data card are denoted within the program as Xk where k = 1, 2, 3, ..., 8. When a transformation is applied to Xk, the transformed variable is denoted as Tk. Tk can be transformed to a new variable which is also denoted as Tk.

The variable to be transformed by a particular transformation as specified on a transformation card is determined by in which of the eight 10-column fields the transformation symbol occurs. The symbol may be punched anywhere within the proper 10-column field. For example, in Fig. 2, four observations of three variables X1, X2, and X3, are read in. The first transformation card instructs the computer to divide X1 by 10.0, add 5.0 to X2, and subtract 60.0 from X3. This results in the transformed variables:

T1	T2	T3
1.0	55.0	30.0
2.0	65.0	40.0
3.0	75.0	50.0
4.0	85.0	60.0

The second transformation card instructs the computer to subtract T3 from T2. This gives the transformed variables:

Table of transformations allowed in MATDS.

Index No.*	Symbol	Meaning
1	LN	natural log
2	LOG	base 10 log
3	SIN	sin
4	COS	cos
5	EXP	e.g. EXP punched in columns 14-16 means e^{T2}
6	+Ti	add i-th transformed variable
7	+Xi	add i-th original (read in) variable
8	+p	add p (p = constant or Ai where Ai = i-th A value)
9	-Ti	subtract
10	-Xi	subtract
11	-p	subtract
12	*Ti	multiply
13	*Xi	multiply
14	*p	multiply
15	ABS	absolute value
16	/Ti	divide
17	/Xi	divide
18	/p	divide
19	=Ti	set equal to
20	=Xi	set equal to
21	=p	set equal to
22	$Ti	raise to power
23	$Xi	raise to power
24	$p	raise to power
25	CUM	each element = sum of itself and all before
26	DELTA	each element = increment from previous element
27	MEAN	each element is mean of this element and preceeding element.
28	ORDER	e.g., if punched in Columns 22-26, rearranges transformed variables and original variables such that the new T3 would be in ascending order.
29	JANET	variable subroutines written for special transformations not included in 28 above transformations.
30	LEANN	
31	SUZIE	

Xi = ith column of data read in
Ti = ith column of the current transformed variable.
p = either a signed constant
or Ai
Ai = ith A value.

*See page 20.

80 COLUMN DATA SHEET

PROGRAM TRANSFORMATION EXAMPLE | JOB NO. | BY | DATE

```
1        10        20        30        40        50        60        70        80
   10.0      50.0      90.0
   20.0      60.0     100.0
   30.0      70.0     110.0
   40.0      80.0     120.0
T  /10.0     +5.0      -6.0
T  -T3
```

Fig. 2. Transformation example.

80 COLUMN DATA SHEET

PROGRAM TRANSFORMATION EXAMPLE | JOB NO. | BY | DATE

```
1         10        20        30        40        50        60        70        80
1.0,.0    5.0,.0    9.0,.0
2.0,.0    6.0,.0    10.0,.0
3.0,.0    7.0,.0    11.0,.0
4.0,.0    8.0,.0    12.0,.0
T1 /1.0,.0    +5.0,     -6.0,.0
T1        -T3
```

Fig. 2. Transformation example.

T1	T2	T3
1.0	25.0	30.0
2.0	25.0	40.0
3.0	25.0	50.0
4.0	25.0	60.0

Transformations JANET, LEANN, and SUZIE are for special transformations. These are variable subroutines which can be written by users familiar with the FORTRAN computer language to perform a transformation which is not included in the Table of Transformations, page 7. These variable subroutines allow the user to perform any desired transformation of an original variable, Xk, or previously transformed variable, Tk. Up to three special transformations can be made, one for each of the three variable subroutines.

Most FORTRAN systems contain standard library functions, such as arc sine, hyperbolic sine, etc., which have not been included in the list of MAIDS transformations. These may be used in a variable subroutine. For example, suppose it is desired to transform the variable T2 as follows:

T2 = T2 * arctan (A3/X7).

This can be done by punching JANET in the second 10-column field of a transformation card (T in Column 1) and including the following subroutine with the MAIDS program subroutines.

```
      SUBROUTINE JANET (J)
      COMMON B(10,10),X(8,400),T(8,409),NM(80),JT(100),
     1IT(100),PT(100),IXT(100),A(8),MN(75),NOOBS,
     2NOVAR, LEE, KFLAG, KAN(50)
      DO 1 I = 1, NOOBS
    1 T(J,I) = T(J,I) * ATAN(A(3)/X(7,I))
      RETURN
      END
```

The SUBROUTINE and COMMON statements in the example above are similar for any variable subroutine (JANET, LEANN, SUZIE), except the name of the subroutine changes. The value of the variable J (two in this example) is the number of the 10-column field that JANET was punched in on the transformation card. Note that JANET was punched in the second 10-column field since the second variable, T2, is to be transformed. The first subscript of T (or X) is the variable number (1 to 8) and the second is the element number or observation number, i.e., T (2, 10) would be the tenth observation of T2. A(3) is the number read in from the third 10-column field of the A card. In the COMMON statement of the subroutines, B, NM, JT, IT, PT, IXT, MN, and LEE are arrays used in other parts of MAIDS and are included here as dummy variables to align COMMON storage. NOOBS is the number of observations of variables read in and NOVAR is the number of variables, ranging from one to eight. KFLAG is an error indicator. Its usual value is one but it is assigned a value of two if an error is encountered. Therefore, if KFLAG has a value of two when control is returned from a subroutine to the main program, an error statement is printed as follows: ** UNIDENTIFIABLE CHARACTER **. KAN is the KAN array from the first input card and ATAN is the FORTRAN library function for computing arc tangent.

Whenever a variable subroutine is used, the corresponding previous variable subroutine having the same name must be removed from the program.

This is necessary because there cannot be two subroutines with the same name in the same program. For some FORTRAN compilers it is required that every subroutine which is referenced in the program be included. Therefore, even if a variable subroutine is not used it should nevertheless be included in the program as a dummy subroutine that actually does nothing. For example:

```
      SUBROUTINE JANET (J)
      COMMON B(10,10), X(8,400), T(8,409), NM(80), JT(100),
     1IT(100), PT(100), IXT(100), A(8), MN(75), NOOBS,
     2NOVAR, LEE, KFLAG, KAN(50)
      DUMMY = 1
      RETURN
      END
```

V* Card

A card with V* punched in Columns 1 and 2 respectively and a number, i (i = 1 to 8), punched in anywhere within Columns 3 and 10 specifies the number of variables to use when a PRINT, REGRESSION, or PACOCO statement is executed. If not specified, the number of variables is taken as eight.

PRINT Card

A card with PRINT punched in Columns 1 to 5 instructs the computer to print out the values of the first i transformed variables (T1, T2, T3, ..., Ti). The value of i is specified on the V* card.

REGRESSION Card

A card with REGRESSION punched in Columns 1 to 10 instructs the computer to determine the regression coefficients of the linear model,

$$T1 = B1 + B2 * T2 + B3 * T3 + \ldots + Bi * Ti.$$

The method of least squares is used to determine the regression coefficients, B1, B2, B3, ..., Bi. Included in the output are the transformed variables and the partial correlation coefficients. The computed (by the regression equation) and the observed values of the dependent variable (T1) are printed along with the standard error of estimate and multiple correlation coefficient.

PACOCO Card

A card with PACOCO punched in Columns 1 to 6 instructs the computer to determine and print the partial correlation coefficients for the first i (as specified on a V* card) transformed variables. As noted above, the partial correlation coefficients are printed when a REGRESSION card is executed; however, a regression is not performed when a PACOCO card is executed. Regression coefficients, B1, B2, ..., Bn, are not determined.

***** Card and *DATA Card

A card with ***** punched in the first five columns instructs the computer to perform the specified operations on the group of data read in prior to the ***** card. The same analysis can be performed on any number of data groups. Each data group must be followed by a ***** card.

The specified operations are contained on the operation cards (T, V*, PRINT, REGRESSION, and PACOCO cards). The operation cards are included with the first group of data and need not be repeated for additional data groups on which the same operations are to be performed.

A *DATA card is used when it is desired to change only the values of constants read in via a card with A punched in the first column. An A card containing a new set of constants should follow the *DATA card. This instructs the computer to perform the specified operations on the original set of data using the new constants. An example problem using the *DATA card will be given later.

END Card

END punched in the first three columns of a card instructs the computer to clear its memory of operation statements. This card is used to separate the series of data groups on which different operations are to be performed. An END card is followed by another set of operation cards (T, V*, PRINT, REGRESSION, and PACOCO) and corresponding data groups.

STOP Card

Whenever a card with STOP punched in the first four columns is encountered, the computer run is terminated. This should be the last card to be read by the computer.

FORTRAN LISTING

MAIDS is written in FORTRAN IV computer language[1,2] for use with the IBM 360/65 computer system at Iowa State University. The FORTRAN listing for the program is given in this section. Since the symbols used in FORMAT statements vary with different computer systems, some of the symbols used with the Iowa State University computer system are explained below.

Symbol	Meaning
'.....'	:All information contained within apostrophes is written on the output sheet in the same form as it is written in the FORMAT statement.
	The first column of each line on the output sheet is reserved for carriage control of the printer. The instruction to leave the first column blank instructs the printer to skip to the next line, (single space), 0(i.e. '0') instructs the printer to skip a line, (double space), and 1 (i.e. '1') instructs the printer to skip to the top of the next output sheet.
X	:This instructs the printer to leave a space blank. For example, 7X instructs the printer to skip seven columns or spaces.
/	:A slash instructs the printer to skip to the next line. Therefore, //// would tell the printer to skip four lines or leave three lines blank.

A complete explanation of the FORTRAN listing for MAIDS would be very lengthy. Therefore, only a brief explanation of the basic purposes of the main program and each subroutine is given. Figure 3 is a schematic diagram showing the relationships between the various subroutines of MAIDS. The arrows in Fig. 3 point to the subroutine which is called. Input subroutines are herein defined as subroutines which are used for reading and interpreting the input cards. Operation subroutines are defined as

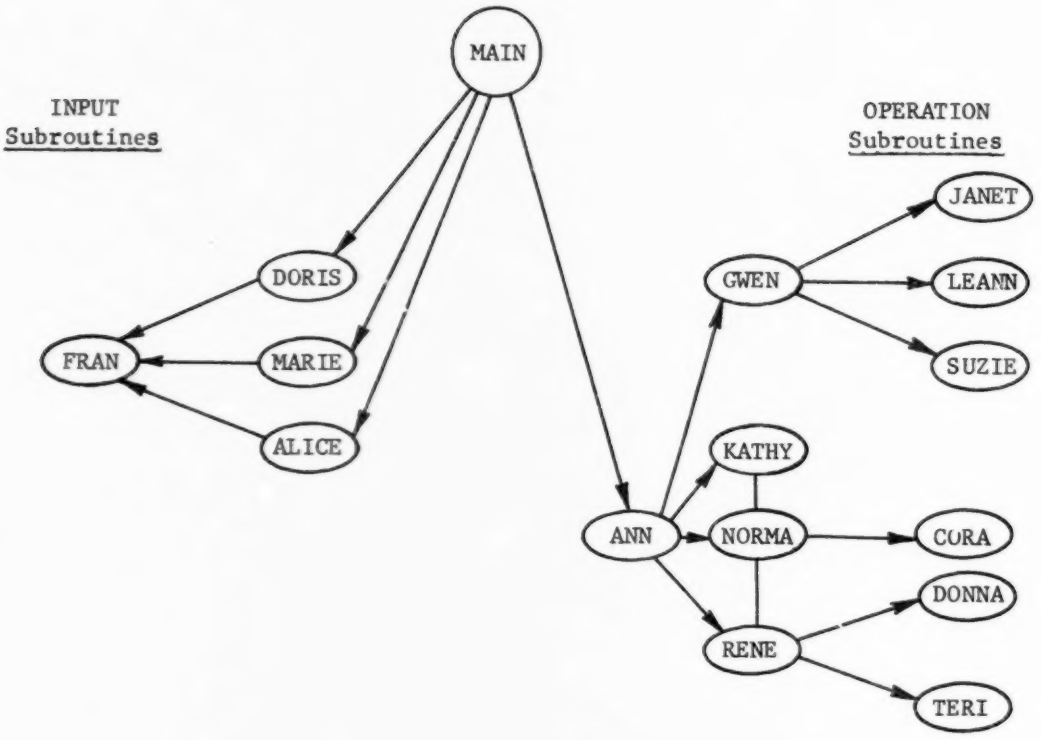

Fig. 3. Schematic diagram of MAIDS subroutines.

subroutines used to perform the specified operations determined by the input subroutines.

MAIN Program

The first card in the data deck is read and the information on it is stored in the KAN array. The succeeding card is then read and the character in the first column is determined by comparison to the KAN array. Control is then transferred to an appropriate subroutine or statement number. For example, if the first character on the card is T, (TRANSFORMATION card), control is transferred to subroutine ALICE which determines what transformation is to be performed. The first character on each input card must correspond to one of the input card forms, i.e., the first character must

```
FORTRAN IV G LEVEL 1, MOD 0             MAIN                DATE = 68044        13/53/59

0001                COMMON B(10,10),X(8,400),T(8,400),NM(80),JT(100),IT(100),BT(100),
                   1IXT(100),A(8),MN(75),NCORS,NOVAR,LEE,KELAC,KAN(50),LAY,FIELD(4,6),
                   1X8(4,5),YA(4,5),LA(4,5),MA(4,5)
0002                READ(1,100) (KAN(K),K=1,50)
0003              100 FORMAT(80A1)
0004                1 LEE=0
0005                  LAY=0
0006                  DO 2 I=1,8
0007                2 A(I)=0.0
0008                  NOVAR=8
0009                  NOORS=0
0010                  KELAC=1
0011                  WRITE(3,200)
0012              200 FORMAT('1',' M A I D S ',/,' DILLINGHAM 1964--RESETT 1966--PRICES
                   11967',///)
0013                3 READ(1,100) (NM(I),I=1,80)
0014                  WRITE(3,300) (NM(I),I=1,80)
0015              300 FORMAT('0',80A1)
0016                  DO 4 K=1,14
0017                  IF(NM(1)-KAN(K)) 4,5,4
0018                4 CONTINUE
0019                  GO TO 22
0020                5 GO TO (3,3,6,7,8,9,16,17,27,1,19,20,21,5),K
0021                6 CALL CORIS
0022                  GO TO (3,24),KELAC
0023                7 CALL ALICE
0024                  GO TO (3,24),KELAC
0025                8 LEE=LEE+1
0026                  IF(NM(2)-KAN(6)) 10,9,10
0027                9 IXT(LEE)=-6
0028                  GO TO 3
0029               10 IF(NM(2)-KAN(9)) 12,11,12
0030               11 IXT(LEE)=-9
0031                  GO TO 3
0032               12 IF(NM(2)-KAN(50)) 24,13,24
0033               13 IXT(LEE)=-4
0034               14 DO 15 KK=1,75
0035               15 MN(KK)=NM(KK+5)
0036                  GO TO 3
0037               16 CALL MARIE
0038                  GO TO (3,24),KELAC
0039               17 LEE=LEE+1
0040                  IXT(LEE)=0
0041                  GO TO 3
0042               19 GO TO 24
0043               20 GO TO 24
0044               21 GO TO 24
0045               22 IF(NM(1)-KAN(31)) 24,23,24
0046               23 CALL ANN
0047                  GO TO 3
0048               24 WRITE(3,400) (NM(I),I=1,80)
0049              400 FORMAT('0',5X,'**UNIDENTIFIABLE CHARACTER**',80A1)
0050               25 READ(1,100) K
0051                  IF(K-KAN(10)) 26,1,26
0052               26 IF(K-KAN(0)) 25,27,25
0053               27 STOP
0054                  END
```

be either C, L, blank, A, T, P, R, V, *, E, or S. If any other character is punched in the first column of a card, the error statement "UNIDENTIFIABLE CHARACTER" is printed. If an unidentifiable character is read, the computer will continue reading cards until either an END or a STOP card is read.

Subroutine FRAN

Subroutine FRAN is used by subroutines DORIS, MARIE, and ALICE to read numerical data. When a card is read by the MAIN program, each character is stored in the array NM. For example, suppose 1.24 is punched in Columns 14 to 17 of a data card. This would not be read in as the number 1.24, but as individual characters: NM(14) = 1, NM(15) = ., NM(16) = 2, and NM(17) = 4. Subroutine FRAN is used to combine these individual characters into one numerical value. The KAN array is used for comparison to determine the numerical value of individual characters. The argument P is the number that is determined, N is the number of the column in which either a plus sign, a minus sign, or the first digit of the number if punched in, and L is the number of the last column in the 10-column field containing the number.

MAIDS could have been written to read numerical values directly. However, the method used allows each input card to be read under the same format. This increases the simplicity of the input card forms and the versatility of the program.

```
FORTRAN IV G LEVEL 1, MOD 0            FRAN            DATE = 68044        13/53/59

0001                SUBROUTINE FRAN(P,A,L)
0002                COMMON B(10,10),X(8,400),T(8,409),NM(80),JT(100),IT(100),PT(100),
                   1IXT(100),A(8),MA(75),NOOBS,NOVAR,LEE,KFLAG,KAN(50)
0003                SN=1.0
0004                NUMBR=0
0005                KPT=-20
0006                IF(N-L) 1,1,3
0007              3 KFLAG=2
0008                RETURN
0009              1 DO 10 I=A,L
0010                J=KAN(I)
0011                DO 2 K=14,27
0012                IF(J-KAN(K)) 2,4,2
0013              2 CONTINUE
0014                GO TO 3
0015              4 K=K-18
0016                IF(K) 5,9,9
0017              5 K=-K
0018                GO TO (6,8,10,10),K
0019              6 IF(KPT) 7,3,3
0020              7 KPT=0
0021                GO TO 10
0022              8 SN=-1.0
0023                GO TO 10
0024              9 NUMBR=NUMBR*10+K
0025                KPT=KPT+1
0026             10 CONTINUE
0027                IF(KPT) 11,11,12
0028             11 D=1.0
0029                GO TO 13
0030             12 D=FLOAT(10**KPT)
0031             13 P=SN*FLOAT(NUMBR)/D
0032                RETURN
0033                END
```

Subroutine DORIS

Control is transferred to subroutine DORIS whenever a data card (blank or A in first column) is read by the MAIN program. If the first column is blank, the number of observations of variables (NOOBS) is increased by one. Each successive column of the data card is checked until a column that is not blank is found. The number of this column, (K), is used to calculate the number of the 10-column field (J) and the number of the last column in the field (L) in which Column K is located. Subroutine FRAN is then used to determine the numerical value of the observed variable X(J,NOOBS), or constant, A(J). This process continues

until the data in each field (J = 1, 2, 3, ..., 8) have been determined. Note that a blank field is not interpreted as zero but is left blank. Also, the values of observed variables are stored in both of the two dimensional arrays, X(J,NOOBS) and T(J,NOOBS). When a transformation is applied to an original variable, Xk (k = 1, 2, 3, ... 8), or a transformed variable, Tk, the new transformed value is stored in the T array. Thus, the values of Xk stored in the X array remain equal to the original variables read in.

```
FORTRAN IV G LEVEL 1, MOD 0          DORIS          DATE = 68044          17/53/55
0001            SUBROUTINE DORIS
0002            COMMON P(10,10),X(8,400),T(8,400),NM(80),JT(100),IT(100),BT(100),
               1IXT(100),A(8),MA(75),NOOBS,NOVAR,LEE,KELAC,KAN(50)
0003            L=2
0004            IF(NM(1)-KAN(6)) 1,2,1
0005          1 NOOBS=NOOBS+1
0006          2 IF(L-50) 3,3,5
0007          3 DO 4 K=1,80
0008            IF(NM(K)-KAN(14)) 6,4,6
0009          4 CONTINUE
0010          5 RETURN
0011          6 J=(K+5)/10
0012            L=J*10
0013            CALL FRAN (F,K,L)
0014            L=L+1
0015            IF(NM(1)-KAN(6)) F,7,8
0016          7 A(J)=F
0017            GO TO 2
0018          8 T(J,NOOBS)=F
0019            X(J,NOOBS)=F
0020            GO TO 2
0021            END
```

Subroutine MARIE

Control is transferred to subroutine MARIE from the MAIN program whenever a V* card is read. Subroutine FRAN is then called to determine the numerical value of the number of variables, NOVAR, punched between Columns 3 and 10 of the V* card. The IXT array and variable subscript, LEE, are used to keep track of the sequence of operations to be performed on the data for each job.

```
FORTRAN IV G LEVEL 1, MOD 0          MARIE          DATE = 68044          13/53/59

0001              SUBROUTINE MARIE
0002              COMMON B(10,10),X(8,400),T(8,409),AM(80),JT(100),IT(100),PT(100),
                 1IXT(100),A(8),MN(75),NOOBS,NOVAR,LEE,KFLAG,KAN(50)
0003              CALL FRAN (P,3,10)
0004              NOVAR=P
0005              LEE=LEE+1
0006              IXT(LEE)=-1
0007              PT(LEE)=NOVAR
0008              IF(NOVAR-8) 2,2,1
0009            1 KFLAG=2
0010            2 RETURN
0011              END
```

Subroutine ALICE

Subroutine ALICE is called by the MAIN program whenever a TRANSFORMATION card is read. This subroutine uses the KAN array to determine which transformation is to be applied and sets the variable, INDEX, equal to the index number corresponding to that transformation, (see Table of Transformations, page 7). This index number is used later by subroutine GWEN which performs the actual transformation. For transformations six through 24, subroutine FRAN is used to determine the numerical values (i or p) in the transformation.

```
FORTRAN IV G LEVEL 1, MOD 0          ALICE          DATE = 68044          13/53/59

0001              SUBROUTINE ALICE
0002              COMMON B(10,10),X(8,400),T(8,409),AM(80),JT(100),IT(100),PT(100),
                 1IXT(100),A(8),MN(75),NOOBS,NOVAR,LEE,KFLAG,KAN(50)
0003              P=0
0004              M=2
0005            1 IF(M-50) 2,2,4
0006            2 DO 3 MM=M,80
0007              IF(AM(MM)-KAN(29)) 5,3,5
0008            3 CONTINUE
0009            4 RETURN
0010            5 J=(MM+9)/10
0011              M=J*10
0012              I=AM(MM)
0013              L=AM(MM+1)
0014              ISN=1
0015              DO 6 K=29,43
0016              IF(I-KAN(K)) 6,8,8
0017            6 CONTINUE
0018            7 KFLAG=2
0019              RETURN
0020            8 K=K-28
```

```
0021            GO TO (9,10,11,12,13,14,21,27,31,35,36,37,38,39,40),K
0022          9 INDEX=5
0023            GO TO 15
0024         10 INDEX=8
0025            GO TO 15
0026         11 INDEX=11
0027            GO TO 15
0028         12 INDEX=15
0029            GO TO 15
0030         13 INDEX=18
0031            GO TO 15
0032         14 INDEX=21
0033         15 DO 16 K=45,46
0034            INDEX=INDEX+1
0035            IF(L-KAN(K)) 16,20,16
0036         16 CONTINUE
0037            INDEX=INDEX+1
0038            IF(L-KAN(39)) 17,19,17
0039         17 L=MM+1
0040            ISA=0
0041         18 CALL FRAN (IP,L,M)
0042            GO TO (41),71),KFLAG
0043         19 ISA=-1
0044         20 L=MM+2
0045            GO TO 18
0046         21 IF(L-KAN(42)) 22,24,22
0047         22 IF(L-KAN(38)) 23,25,23
0048         23 IF(L-KAN(48)) 7,24,7
0049         24 INDEX=2
0050            GO TO 41
0051         25 INDEX=39
0052            GO TO 41

0053         26 INDEX=1
0054            GO TO 41
0055         27 IF(L-KAN(40)) 28,25,28
0056         28 IF(L-KAN(50)) 7,30,7
0057         29 INDEX=3
0058            GO TO 41
0059         30 INDEX=31
0060            GO TO 41
0061         31 IF(L-KAN(42)) 32,33,32
0062         32 IF(L-KAN(50)) 7,34,7
0063         33 INDEX=4
0064            GO TO 41
0065         34 INDEX=25
0066            GO TO 41
0067         35 INDEX=5
0068            GO TO 41
0069         36 INDEX=15
0070            GO TO 41
0071         37 INDEX=26
0072            GO TO 41
0073         38 INDEX=27
0074            GO TO 41
0075         39 INDEX=28
0076            GO TO 41
```

```
0077          40 INDEX=2S
0078          41 LEE=LEE+1
                 JT(LEE)=J
0080             PT(LEE)=P
0081             IT(LEE)=ISN*IFIX(P)
0082             IF(IABS(IT(LEE))-8) 42,43,42
0083          42 IF(L-MM-1) 43,43,7
0084          43 IXT(LEE)=INDEX
0085             M=M+1
0086             GO TO 1
0087             END
```

Subroutine ANN

Subroutine ANN is called by the MAIN program whenever a ***** card or *DATA card is read. This subroutine then instructs the other operation subroutines to perform the desired sequence of operations.

```
FORTRAN IV G LEVEL 1, MOD 0              ANN                DATE = 68044          12/53/55

0001              SUBROUTINE ANN
0002              COMMON P(10,10),X(8,400),T(9,409),NM(80),JT(100),IT(100),PT(100),
                 1IXT(100),A(8),MN(75),ACCBS,ACVAR,LEE,KELAC,KAN(50),LAY
0003              LAY=0
0004              DO 13 K=1,LEE
0005              IF(IXT(K)) 2,1,9
0006            1 CALL RENE
0007              GO TO 13
0008            2 L=-IXT(K)
0009              GO TO (3,4,5,6,7,8),L
0010            3 ACVAR=PT(K)
0011            4 GO TO 13
0012            5 NOOBS=PT(K)
0013            6 GO TO 13
0014            7 CALL KATHY
0015              GO TO 13
0016            8 CALL ACRAZ
0017              GO TO 13
0018            9 J=JT(K)
0019              I=IT(K)
0020              P=PT(K)
0021              IX=IXT(K)
0022              IF(I) 10,12,12
0023           10 I=-I
0024              P=A(I)
0025           11 I=-I
0026           12 CALL GWEN (J,I,P,IX)
0027           13 CONTINUE
0028              LAY=0
0029              IF(NM(2)-KAN(1)) 15,14,15
0030           14 LEE=0
0031              GO TO 17
0032           15 IF(NM(2)-KAN(3)) 16,17,16
0033           16 ACCBS=0
0034           17 WRITE(3,100)
0035          100 FORMAT('1',10X,'DATA')
0036              RETURN
0037              END
```

Subroutine GWEN

This subroutine performs the transformations specified on the transformation cards. Special transformations can be performed using the variable subroutines JANET, LEANN, and SUZIE.

```
FORTRAN IV G LEVEL 1, MOD 0             GWEN              DATE = 68044          13/53/59

0001                SUBROUTINE GWEN (J,I,P,IX)
0002                COMMON A(10,10),X(P,400),T(8,409),NM(80),JT(100),IT(100),PT(100),
                   1IXT(100),A(8),MN(75),NOOBS,NOVAR,LEE,KFLAG,KAN(50)
0003                GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23
                   1,24,25,26,27,28,29,30,31),IX
0004              1 DO 101 K=1,NOOBS
0005                IF(T(J,K)) 1000,1000,101
0006            101 T(J,K)=ALOG(T(J,K))
0007                RETURN
0008              2 DO 102 K=1,NOOBS
0009                IF(T(J,K)) 1000,1000,102
0010            102 T(J,K)=ALOG10(T(J,K))
0011                RETURN
0012           1000 WRITE(3,1001)
0013           1001 FORMAT('0',10X,'****ARGUMENT NEGATIVE OR ZERO****')
0014                RETURN
0015              3 DO 103 K=1,NOOBS
0016            103 T(J,K)=SIN(T(J,K))
0017                RETURN
0018              4 DO 104 K=1,NOOBS
0019            104 T(J,K)=COS(T(J,K))
0020                RETURN
0021              5 DO 105 K=1,NOOBS
0022            105 T(J,K)=EXP(T(J,K))
0023                RETURN
0024              6 DO 106 K=1,NOOBS
0025            106 T(J,K)=T(J,K)+T(I,K)
0026                RETURN
0027              7 DO 107 K=1,NOOBS
0028            107 T(J,K)=T(J,K)+X(I,K)
0029                RETURN
0030              8 DO 108 K=1,NOOBS
0031            108 T(J,K)=T(J,K)+P
0032                RETURN
0033              9 DO 109 K=1,NOOBS
0034            109 T(J,K)=T(J,K)-T(I,K)
0035                RETURN
0036             10 DO 110 K=1,NOOBS
0037            110 T(J,K)=T(J,K)-X(I,K)
0038                RETURN
0039             11 DO 111 K=1,NOOBS
0040            111 T(J,K)=T(J,K)-P
0041                RETURN
0042             12 DO 112 K=1,NOOBS
0043            112 T(J,K)=T(J,K)*T(I,K)
0044                RETURN
0045             13 DO 113 K=1,NOOBS
0046            113 T(J,K)=T(J,K)*X(I,K)
0047                RETURN
0048             14 DO 114 K=1,NOOBS
0049            114 T(J,K)=T(J,K)*P
0050                RETURN
0051             15 DO 115 K=1,NOOBS
```

```
0052          115 T(J,K)=ABS(T(J,K))
0053              RETURN
0054           16 DO 116 K=1,ACCRS
0055              IF(T(I,K)) 116,1000,116
0056          116 T(J,K)=T(J,K)/T(I,K)
0057              RETURN
0058           17 DO 117 K=1,NOOBS
0059              IF(X(I,K)) 117,1000,117
0060          117 T(J,K)=T(J,K)/X(I,K)
0061              RETURN
0062           18 IF(P) 118,1000,118
0063          118 DO 218 K=1,NOOBS
0064          218 T(J,K)=T(J,K)/F
0065              RETURN
0066           19 DO 119 K=1,NOOBS
0067          119 T(J,K)=T(I,K)
0068              RETURN
0069           20 DO 120 K=1,NOOBS
0070          120 T(J,K)=X(I,K)
0071              RETURN
0072           21 DO 121 K=1,NOOBS
0073          121 T(J,K)=P
0074              RETURN
0075           22 DO 122 K=1,NOOBS
0076              IF(T(J,K)) 1000,122,122
0077          122 T(J,K)=T(J,K)**T(I,K)
0078              RETURN
0079           23 DO 123 K=1,ACCRS
0080              IF(T(J,K)) 1000,123,123
0081          123 T(J,K)=T(J,K)**X(I,K)
0082              RETURN
0083           24 KP=P
0084              IF(P-FLOAT(KP)) 324,124,324
0085          124 DO 224 K=1,ACCRS
0086          224 T(J,K)=T(J,K)**KP
0087              RETURN
0088          324 DO 424 K=1,ACCRS
0089              IF(T(J,K)) 1000,424,424
0090          424 T(J,K)=T(J,K)**P
0091              RETURN
0092           25 DO 125 K=2,NOOBS
0093          125 T(J,K)=T(J,K)+T(J,K-1)
0094              RETURN
0095           26 K=NOOBS
0096          126 T(J,K)=T(J,K)-T(J,K-1)
0097              K=K-1
0098              IF(K-1) 226,226,126
0099          226 T(J,1)=0.0
0100              RETURN
0101           27 KP=NOOBS-1
0102              DO 127 K=1,KP
0103              I=NOOBS-K-1
0104          127 T(J,I)=(T(J,I)+T(J,I-1))/2.0
```

```
0105              T(J,1)=0.0
0106              RETURN
0107           29 HUGE=10.0**38.0
0108              KP=NOOBS-1
0109              DO 328 I=1,KP
0110              P=HUGE
0111              DO 228 L=I,NOOBS
0112              IF(P-T(J,L)) 228,228,128
0113          128 P=T(J,L)
0114              K=L
0115          228 CONTINUE
0116              DO 328 L=1,8
0117              P=X(L,I)
0118              X(L,I)=X(L,K)
0119              X(L,K)=P
0120              P=T(L,I)
0121              T(L,I)=T(L,K)
0122          328 T(L,K)=P
0123              RETURN
0124           29 CALL JANET (J)
0125              RETURN
0126           30 CALL LEANN (J)
0127              RETURN
0128           31 CALL SUZIE (J)
0129              RETURN
0130              END
```

Variable subroutines JANET, LEANN, and SUZIE are included here as dummy subroutines (see page 11).

```
FORTRAN IV G LEVEL 1, MOD 0          JANET          DATE = 68044          12/53/59

0001              SUBROUTINE JANET (J)
0002              COMMON B(10,10),X(8,400),T(8,400),NM(80),JT(100),IT(100),PT(100),
                 1IXT(100),A(8),MN(75),NOOBS,NOVAR,LEE,KFLAG,KAN(50)
0003              DUMMY=1
0004              RETURN
0005              END
```

```
FORTRAN IV G LEVEL 1, MOD 0             LEANN               DATE = 68044          13/53/59

0001                SUBROUTINE LEANN (J)
0002                COMMON B(10,10),X(8,400),T(8,409),NM(80),JT(100),IT(100),PT(100),
                   1IXT(100),A(8),MM(75),NOOBS,NOVAR,LEE,KFLAG,KAN(50)
0003                DUMMY=1
0004                RETURN
0005                END
```

```
FORTRAN IV G LEVEL 1, MOD 0             SUZIE               DATE = 68044          13/53/59

0001                SUBROUTINE SUZIE (J)
0002                COMMON B(10,10),X(8,400),T(8,409),NM(80),JT(100),IT(100),PT(100),
                   1IXT(100),A(8),MM(75),NOOBS,NOVAR,LEE,KFLAG,KAN(50)
0003                DUMMY=1
0004                RETURN
0005                END
```

Subroutine KATHY

Subroutine KATHY prints out the current values of the transformed variables and the constants in the A array. This subroutine is used whenever a PRINT card is executed, and by subroutine NORMA when PACOCO or REGRESSION cards are executed.

```
FORTRAN IV G LEVEL 1, MOD 0             KATHY               DATE = 68044          13/53/55

0001                SUBROUTINE KATHY
0002                COMMON B(10,10),X(8,400),T(8,409),NM(80),JT(100),IT(100),PT(100),
                   1IXT(100),A(8),MM(75),NOOBS,NOVAR,LEE,KFLAG,KAN(50)
0003                WRITE(3,100) (I,I=1,NOVAR)
0004              100 FORMAT('1',' TRANSFORMED VARIABLES',//,'0  OBS',8(13X,'I',I1))
0005                DO 1 I=1,NOOBS
0006                  1 WRITE(3,200) I,(T(J,I),J=1,NOVAR)
0007              200 FORMAT(1X,I5,8E15.6)
0008                WRITE(3,300) (I,A(I),I=1,8)
0009              300 FORMAT('0',/,10X,'A VALUES',8(/,20X,I1,')',E15.6))
0010                RETURN
0011                END
```

Subroutine NORMA

Using the previously defined notation where T1 is the dependent variable and T2, T3, ..., Tn are independent variables, (n = number of variables, two to eight), subroutine NORMA determines the rectangular systems matrix, A/C, for subsequent use in the regression analysis. This matrix contains the normal equations used in the least squares method of estimating regression coefficients[3]. The first n columns contain the matrix A and the last column contains the matrix or vector C.

NOOBS	$\Sigma T2$	$\Sigma T3$	.	.	ΣTn	$\Sigma T1$
$\Sigma T2$	$\Sigma(T2)^2$	$\Sigma T2T3$	.	.	$\Sigma T2Tn$	$\Sigma T2T1$
$\Sigma T3$	$\Sigma T2T3$	$\Sigma(T3)^2$	.	.	$\Sigma T3Tn$	$\Sigma T3T1$
.	.	.	.	.	.	.
.	.	.	.	.	.	.
ΣTn	$\Sigma T2Tn$	$\Sigma T3Tn$	.	.	$\Sigma(Tn)^2$	$\Sigma TnT1$

NOOBS is the number of observations of the variables.

```
FORTRAN IV G LEVEL 1, MOD 0             NORMA           DATE = 68044        13/53/59

0001                SUBROUTINE NORMA
0002                COMMON B(10,10),X(8,400),T(8,409),NM(80),JT(100),IT(100),PT(100),
                   1IXT(100),A(8),MN(75),NOOBS,NOVAR,LEE,KFLAG,KAN(50)
0003                DO 1 I=1,10
0004                DO 1 J=1,10
0005              1 B(I,J)=0.0
0006                N=NOVAR-1
0007                DO 5 M=1,NOOBS
0008                DO 5 I=1,NOVAR
0009                FACTR=T(I,M)
0010                IF(I-1) 3,2,3
0011              2 FACTR=1.0
0012              3 J=I
0013             31 J=J+1
0014                IF(J-NOVAR) 4,4,41
0015              4 B(I,J)=B(I,J)+FACTR*T(J,M)
0016                GO TO 31
0017             41 B(I,NOVAR+1)=FACTR*T(I,M)+B(I,NOVAR+1)
0018              5 B(I,I)=T(I,M)*T(I,M)+B(I,I)
0019                B(1,1)=NOOBS
0020                DO 6 I=1,N
0021                K=I+1
0022                DO 6 J=K,NOVAR
0023              6 B(J,I)=B(I,J)
0024                CALL KATHY
0025                CALL CORR (NOVAR)
0026                RETURN
0027                END
```

Subroutine CORA

Subroutine CORA is called by subroutine NORMA to calculate the partial correlation coefficients between all possible pairs of variables. These are calculated using the equation[4]

$$r_{ij} = \frac{N\Sigma TiTj - \Sigma Ti\Sigma Tj}{\sqrt{[N\Sigma Ti^2 - (\Sigma Ti)^2][N\Sigma Tj^2 - (\Sigma Tj)^2]}}$$

where, r_{ij} = partial correlation coefficient between variables Ti and Tj

N = number of observations of variables or NOOBS

i,j = integer between 1 and the number of variables (maximum of 8).

More specifically, these are zero order partial correlation coefficients; the correlation between any two variables is determined neglecting all other variables. This is opposed to higher order partial correlations which determine the correlation between variables while holding all other variables constant.

```
FORTRAN IV G LEVEL 1, MOD 0             CORA                DATE = 68044          17/53/55

0001                SUBROUTINE CORA (A)
0002                COMMON B(100),X(8,400),T(3272)
0003                WRITE(3,100)
0004            100 FORMAT('1',22X,'PARTIAL CORRELATION COEFFICIENTS')
0005                M=B(1)*8.0
0006                SYY=0.0
0007                DO 1 I=1,M,8
0008              1 SYY=SYY+T(I)*T(I)
0009                NN=N*10+1
0010                SYY=B(1)*SYY-B(NN)*B(NN)
0011                DO 4 I=2,N
0012                K=I-1
0013                M=K*10+1
0014                II=NN+I-1
0015                B(91)=(B(1)*B(II)-B(NN)*B(I))/SQRT(SYY*(B(1)*B(M)-B(I)*B(I)))
0016                IF(I-2) 4,4,2
0017              2 DO 3 J=2,K
0018                KK=(J-1)*10+J
0019                L=M+J-1
0020              3 B(J+90)=(B(1)*B(L)-B(J)*B(I))/SQRT((B(1)*B(M)-B(I)*B(I))*(B(1)*
                   1B(KK)-B(J)**2))
0021              4 WRITE(3,200) I,(B(J+90),J=1,K)
0022            200 FORMAT('0',1X,I8,F10.3,6F8.3)
0023                WRITE(3,300) (I,I=1,K)
0024            300 FORMAT('0',1X,I17,6I8)
0025                RETURN
0026                END
```

Subroutine RENE

This subroutine is called whenever a REGRESSION card is executed. Subroutine RENE first calls subroutine NORMA to determine the rectangular systems matrix and partial correlation coefficients. Subroutine DONNA is then called to calculate the coefficients (B1, B2, ..., Bn) of the regression equation,

$$T1 = B1 + B2*T2 + B3*T3 + \ldots + Bn*Tn.$$

The resulting regression equation is then used for each observation to calculate the estimated value, $\hat{T1}$, of the observed dependent variable, T1, and the difference, $T1 - \hat{T1}$, between observed and calculated values. This difference is also expressed as a percentage of the average of the observed and calculated values and as a percentage of the mean of all observed values of the dependent variable. The following quantities are also determined[3].

$$\text{Root mean square of percent differences} = \sqrt{\frac{\Sigma(\%\ \text{difference})^2}{N - n}}$$

$$\text{Standard error of estimate} = \sqrt{\frac{\Sigma(T1 - \hat{T1})^2}{N - n}}$$

and,

$$100R = \frac{N\Sigma T1 \cdot \hat{T1} - \Sigma T1 \Sigma \hat{T1}}{\sqrt{[N\Sigma T1^2 - (\Sigma T1)^2][N\Sigma \hat{T1}^2 - (\Sigma \hat{T1})^2]}} \quad (100)$$

where, N = number of observations, NOOBS

n = number of variables, NOVAR

100R = multiple correlation coefficient expressed in percent.

Finally, subroutine TERI is called to print out the matrices used in calculating the regression coefficients.

```
FORTRAN IV G LEVEL 1, MOD 0          REAR                DATE = 68044          13/53/59

0001                  SUBROUTINE REAR
0002                  COMMON B(10,10),X(8,400),T(8,400),AA(80),JT(100),IT(100),PT(100),
                     1IXT(100),A(8),MN(75),NOOBS,NOVAR,LEE,KFLAG,KAA(50)
0003                  CALL ACRRA
0004                  N=NOVAR
0005                  K=NOVAR+1
0006                  DO 1 J=1,K
0007                  JJ=J+400
0008                  DO 1 I=1,N
0009                1 T(I,JJ)=B(I,J)
0010                  WRITE(3,100)
0011                100 FORMAT(//////////,1X,'REGRESSION MODEL ----- Y1 = B1 + B2*X2 + B3*X
                     13 + . . . . . + BN*XN',//)
0012                  CALL DCRRA (N,1)
0013                  WRITE(3,200) (I,B(I,K),I=1,N)
0014                200 FORMAT('0',17X,'B',I1,'=',E20.8)
0015                  WRITE(3,300)
0016                300 FORMAT('1',2X,'OBS',3X,'OBSERVED Y1',3X,'COMPUTED Y1',5X,'D  I  F
                     1 F  E  R  E  N  C  E',/,40X,'ACTUAL',9X,'Y/Y',8X,'Z')
0017                  Z=0.0
0018                  DO 2 K=1,NOOBS
0019                2 Z=Z+T(1,K)
0020                  YBAR=Z/(FLOAT(NOOBS))
0021                  Z=0.0
0022                  SY=0.0
0023                  SYH=0.0
0024                  SYY=0.0
0025                  SYHYH=0.0
0026                  SYYH=0.0
0027                  SDIFF=0.0
0028                  DO 4 K=1,NOOBS
0029                  Y=T(1,K)
0030                  YHAT=B(1,N+1)
0031                  DO 3 I=2,NOVAR
0032                3 YHAT=YHAT+B(I,N+1)*T(I,K)
0033                  T(8,K)=YHAT
0034                  DIFF=Y-YHAT
0035                  SDIFF=SDIFF+DIFF*DIFF
0036                  E=100.0*DIFF/YBAR
0037                  P=200.0*DIFF/(Y+YHAT)
0038                  Z=Z+P*P
0039                  SY=SY+Y
0040                  SYH=SYH+YHAT
0041                  SYY=SYY+Y*Y
0042                  SYHYH=SYHYH+YHAT*YHAT
0043                  SYYH=SYYH+Y*YHAT
0044                4 WRITE(3,400) K,T(1,K),YHAT,DIFF,P,E
0045                400 FORMAT(1X,I5,2E14.5,F14.4,F11.3,F10.3)
0046                  P=ACCRS-N
0047                  SEE=(SDIFF/P)**0.5
0048                  Z=(Z/P)**0.5
0049                  P=ACCRS
0050                  R=(P*SYYH-SY*SYH)/SQRT((P*SYY-SY*SY)*(P*SYHYH-SYH*SYH)*0.000)
0051                  WRITE(3,500) Z,SEE,R
0052                500 FORMAT('0',///,1X,'ROOT MEAN SQUARE OF PERCENT DIFFERENCES',F18.3,
                     1///,1X,'STANDARD ERROR OF ESTIMATE',13X,F18.3,/
                     1///,1X,'R = MULTIPLE CORRELATION COEFFICIENT     100(R) = ',F8.3,//
                     1/,6X,'Z = 100(DIFF)/(MEAN OBSERVED Y1)')
0053                  CALL TERI (N,1)
0054                  RETURN
0055                  END
```

Subroutine DONNA

Subroutine DONNA determines the regression coefficients (B1, B2, ..., Bn) along with the inverted matrix, A^{-1}, of the matrix A found in subroutine NORMA. The invert of matrix A, A^{-1}, is found by using the abbreviated Doolittle method[3,5]. The solution vector, B, containing the regression coefficients, is then equal to the product of matrix A^{-1} and vector C, (from subroutine NORMA).

```
FORTRAN IV G LEVEL 1, MOD 0             DONNA            DATE = 68044        13/53/59

0001                SUBROUTINE DONNA (N,M)
0002                COMMON B(10,10)
0003                NM=N+M
0004                NM1=NM+1
0005                N1=N+1
0006                N2=N+2
0007                DO 1 K=1,N
0008                B(N2,K)=K
0009              1 B(K,NM1)=K
0010                DO 10 IP=1,N
0011                K=N+1-IP
0012                BIG=0.0
0013                DO 4 I=1,K
0014                DO 4 J=1,K
0015                IF(ABS(B(I,J))-BIG) 4,4,3
0016              3 BIG=ABS(B(I,J))
0017                L=I
0018                MM=J
0019              4 CONTINUE
0020                DO 5 I=1,N2
0021                TEMP=B(I,1)
0022                B(I,1)=B(I,MM)
0023              5 B(I,MM)=TEMP
0024                DO 6 J=1,NM1
0025                TEMP=B(1,J)
0026                B(1,J)=B(L,J)
0027              6 B(L,J)=TEMP
0028                B(N1,NM1)=1.0/B(1,1)
0029                DO 7 K=2,NM
0030              7 B(N1,K-1)=B(N1,NM)*B(1,K)
0031                TEMP=B(1,NM1)
0032                BIG=B(N2,1)
0033                DO 9 I=2,N
0034                IM1=I-1
0035                DO 8 J=2,NM
0036              8 B(IM1,J-1)=B(I,J)-B(I,1)*B(N1,J-1)
0037                B(IM1,NM)=-B(I,1)*B(N1,NM)
0038                B(IM1,NM1)=B(I,NM1)
0039              9 B(N2,IM1)=B(N2,I)
0040                B(N,NM1)=TEMP
0041                B(N2,N)=BIG
0042                DO 10 K=1,NM
0043             10 B(N,K)=B(N1,K)
0044                DO 11 K=1,N
0045                TEMP=B(K,NM1)
0046                B(K,NM1)=B(N2,K)
```

```
0047          11 B(N2,K)=TEMP
0048             DO 13 K=1,M
0049             DO 13 I=1,N
0050             TEMP=P(I,1)
0051             DO 12 J=2,NM
0052          12 P(I,J-1)=P(I,J)
0053          13 P(I,NM)=TEMP
0054             MM=N-1
0055             DO 16 I=1,MM
0056             XI=I
0057             DO 14 J=1,N
0058             IF(P(N2,J)-XI) 14,15,14
0059          14 CONTINUE
0060          15 DO 16 K=1,N2
0061             TEMP=P(K,I)
0062             P(K,I)=P(K,J)
0063          16 P(K,J)=TEMP
0064             DO 19 I=1,MM
0065             XI=I
0066             DO 17 J=1,N
0067             IF(P(J,NM1)-XI) 17,18,17
0068          17 CONTINUE
0069          18 DO 19 K=1,NM1
0070             TEMP=P(I,K)
0071             P(I,K)=P(J,K)
0072          19 P(J,K)=TEMP
0073             RETURN
0074             END
```

Subroutine TERI

This subroutine is used to print out the matrices used in the regression analysis. First, the rectangular system's matrix, A/C, as calculated by subroutine NORMA, is printed. The identity or unit matrix, I, is then calculated as the matrix product, $A^{-1}A$. All of the elements of the identity matrix should be zero except for the elements in the principal diagonal which should be one. The identity matrix and solution vector, B, are printed out as matrix I/B. Finally, matrix A^{-1} is printed out with the solution vector.

```
FORTRAN IV G LEVEL 1, MOD 0             TEST                DATE = 68044          12/53/59

0001                SUBROUTINE TEST (N,M)
0002                COMMON B(10,10),X(8,400),T(8,408),NM(80),JT(100),IT(100),PT(100),
                   1IXT(100),A(8),MN(75),NOOBS,NOVAR,LSE,KFLAG,KAN(50)
0003                WRITE(3,100)
0004              100 FORMAT('1',30X,'MATRIX EQUATION',5X,'A(B)=C',///,10X,'RECTANGULAR
                   1SYSTEMS MATRIX',5X,'(A/C)')
0005                K=N+M
0006                L=400+K
0007                DO 1 I=1,N
0008              1 WRITE(3,200) (T(I,J),J=401,L)
0009              200 FORMAT('0',8E13.5)
0010                WRITE(3,300)
0011              300 FORMAT('0',///,' IDENTITY AUGMENTED WITH SOLUTION VECTOR(S)  (I/B)
                   1')
0012                L=N+1
0013                DO 4 I=1,N
0014                DO 3 J=1,K
0015                KK=400+J
0016                C=0.0
0017                DO 2 LL=1,N
0018              2 C=C+B(I,LL)*T(LL,KK)
0019              3 B(L,J)=C
0020              4 WRITE(3,400) (B(L,J),J=1,K)
0021              400 FORMAT('0',8E13.6)
0022                WRITE(3,500)
0023              500 FORMAT('0',///,1X,'INVERTED RSM MATRIX    (A/B)')
0024                DO 5 I=1,N
0025              5 WRITE(3,200) (B(I,J),J=1,K)
0026                WRITE(3,600)
0027              600 FORMAT('1')
0028                RETURN
0029                END
```

EXAMPLE INPUT AND OUTPUT

The major use of MAIDS at Iowa State University has been for the determination of prediction equations for diatomite filter cake resistance indices or β-indices. A log-log plot of β-index versus the ratio of suspended solids concentration of the filter influent, C_S, to body feed concentration of the filter aid, C_D, is a straight line. This corresponds to the equation

$$\beta = a(C_S/C_D)^{b_2}$$

where a and b_2 are constants. If a is replaced by the constant, 10^{b_1}, the equation can be made linear by a logarithmic transformation, i.e., $\log \beta = b_1 + b_2 \log (C_S/C_D)$. MAIDS can therefore be used to determine the coefficients b_1 and b_2. Additional β-index prediction equations of the form,

$$\beta = 10^{b_1} (C_S/C_D)^{b_2} C_D^{b_3}$$

and

$$\beta = 10^{b_1}(C_S/C_D)^{b_2}C_D^{b_3}\xi^{b_4}$$

were presented and analyzed by Dillingham *et al.*[6]. Then, b_1, b_2, b_3, and b_4 are empirical constants, and ξ is the filter aid resistance index or ξ-index.

Two examples of the use of MAIDS are presented below. The first example involves the determination of β-index prediction equations. The second shows the use of MAIDS for problems other than regression analysis.

EXAMPLE 1

Given the filter cake resistance index, β, the body feed concentration, C_D, and the suspended solids concentration of the filter influent, C_S, for each of 18 filter runs, determine the coefficients of the equations

$$\beta = 10^{b_1}(C_S/C_D)^{b_2}$$

and

$$\beta = 10^{b_1}(C_S/C_D)^{b_2}C_D^{b_3} .$$

EXAMPLE 2

Compute the quantities

$$y = a + bX$$

and

$$z = a + bX + cX^2$$

for ten given values of X and for values of a, b, and c of - 2, - 3, and 6 respectively and 5, 7, and - 3 respectively.

Output the results with X in ascending order in Column 1, y in Column 2, and z in Column 3.

The input sheets for these two problems are shown in Fig. 4. Note that the KAN array card must be included as the first card in the input data.

80 COLUMN DATA SHEET

PROGRAM	JOB NO.	BY	DATE
EXAMPLE INPUT		Harold Bridges	

```
1        10        20        30        40        50        60        70        80
CLDTPAVRSEFOM +-.0123456789 +-*/=$LSCEADMOJ TXANIU
C  EXAMPLE NUMBER 1
C  DETERMINATION OF BETA INDEX PREDICTION EQUATIONS
C  DATA FOLLOWS
C  BETA/E06 = BETA INDEX DIVIDED BY 1,000,000
L  RUN NO.      CS        CD        BETA/E06
   1.0         11.3      56.5       70.70
   2.0         13.4     126.7       17.58
   3.0         13.0     166.4        7.986
   4.0         18.2     181.6        8.752
   5.0         14.4     152.2       14.19
   6.0         15.0     230.0        5.775
   7.0         16.0     257.3        4.859
   8.0         13.1     257.3        2.936
   9.0         16.0     146.2        9.521
  10.0         15.5     197.8        5.428
  11.0         15.3     267.7        3.128
  12.0         18.4     290.0        3.188
  13.0         14.6     225.3        4.177
  14.0         15.0     136.2       12.11
  15.0         18.6     107.0       21.28
  16.0         16.6      84.7       38.96
  17.0         17.4      73.7       45.71
  18.0          9.60     105.8      11.68
```

Fig. 4. Example input.

80 COLUMN DATA SHEET

PROGRAM			JOB NO.		BY		DATE
1 10	20	30	40	50	60	70	80
C OPERATIONS TO BE PERFORMED FOLLOW							
T			*1000000				
T =T4							Data can be read in
T LOG	LOG	LOG					any of the available
T	-T3						10-column fields and
V* 2							then shifted as
REGRESSION							desired.
V* 3							
REGRESSION							
*****							
END							
C EXAMPLE NUMBER 2							
T ORDER	=T1	=T1					Operation cards do
T	*A2	$2.0					not necessarily
T	+A1	*A3					have to follow
T		+T2					data cards.
V* 3							
PRINT							
L X VALUES							
6.0							
4.0							
1.0							
9.0							
7.0							

Fig. 4. (Continued)

80 COLUMN DATA SHEET

PROGRAM	JOB NO.	BY	DATE

1 10 20 30 40 50 60 70 80

```
     3.0
     5.0
     8.0
    10.0
     2.0
A   -2.0      3.0      6.0
*DATA
A    5.0      7.0     -3.0
*****
STOP
```


Fig. 4. (Continued)

```
M A I D S
DILLINGHAM 1964--BESETT 1966--BRIDGES 1967

C   EXAMPLE NUMBER 1
C   DETERMINATION OF BETA INDEX PREDICTION EQUATIONS
C   DATA FOLLOWS
C    BETA/E06 = BETA INDEX DIVIDED BY 1,000,000
L RUN NO.      CS          CD       BETA/E06
    1.0       11.2        56.5       20.70
    2.0       13.4       126.7       17.58
    3.0       13.0       166.4        7.986
    4.0       18.2       181.6        8.752
    5.0       14.4       152.2       14.19
    6.0       15.0       230.0       5.775
    7.0       16.0       257.3       4.859
    8.0       13.1       257.3       2.936
    9.0       16.0       146.2       9.521
   10.0       15.5       197.8       5.428
   11.0       15.3       267.7       3.128
   12.0       19.4       250.0       3.188
   13.0       14.6       225.3       4.177
   14.0       15.0       196.2       12.11
   15.0       18.6       107.0       21.28
   16.0       14.6        84.7       38.96
   17.0       17.4        73.7       45.71
   18.0        9.60      105.8       11.68
C   OPERATIONS TO BE PERFORMED FOLLOW
T                                  *1000000
T   =T4
T   LOG        LOG         LOG
T              -T3
V*  2
REGRESSION
V*  3
REGRESSION
*****
```

Reproduced from best available copy.

TRANSFORMED VARIABLES

OBS	T1	T2
1	C.784941E C1	-C.658571E 00
2	0.724502E 01	-0.975672E 00
3	0.690233E 01	-0.110721E 01
4	C.694211E C1	-C.999044E CC
5	0.715198E 01	-0.102405E 01
6	C.676155E 01	-0.118564E 01
7	C.668655E C1	-C.120632E 01
8	0.646776E 01	-0.129317E 01
9	0.697868E 01	-0.960828E 00
10	0.673464E 01	-0.110589E C1
11	0.649527E 01	-0.124296E 01
12	C.650352E C1	-C.119758E 01
13	0.662086E 01	-C.118841E 01
14	0.708314E 01	-0.958087E 00
15	C.732797E C1	-C.759871E CO
16	0.759062E 01	-0.707776E CC
17	0.766C1E 01	-0.626919E 00
18	C.7C6744E 01	-C.1C4221E 01

A VALUES

1)	C.C
2)	C.C
3)	0.0
4)	0.0
5)	C.C
6)	C.C
7)	0.0
8)	0.0

(Author's note:

This output is in exponential form. The number following E is the power of ten.)

PARTIAL CORRELATION COEFFICIENTS

2 0.957

1

(Author's note:

$r_{12} = 0.957$

This is the correlation coefficient between variables one and two.)

REGRESSION MODEL ----- T1 = B1 + B2*T2 + B3*T3 + ... + BN*TN

B1 = 8.98211765

B2= 1.64752510

OBS	OBSERVED T1	COMPUTED T1	D I F F E R E N C E		
			ACTUAL	O/O	Z
1	0.78494E 01	0.76206E 01	C.2288E CC	2.955	3.267
2	C.7245CE C1	C.7C816E 01	0.1634E 00	2.282	2.334
3	0.69023E C1	C.68252E C1	C.765BE-01	1.122	1.099
4	0.69421E 01	0.70360E 01	-0.9394E-01	-1.344	-1.341
5	C.7152CE C1	C.69873E C1	0.1646E 00	2.329	2.351
6	0.67616E C1	C.66726E 01	C.8858E-01	1.325	1.27C
7	C.66865E 01	0.66323E 01	0.5426E-01	0.815	0.775
8	C.64678E C1	C.64631E C1	C.4647E-C2	0.072	0.066
9	0.69787E 01	0.71105E 01	-C.131BE CC	-1.871	-1.882
1C	C.67346E C1	0.68279E 01	-0.9327E-01	-1.375	-1.332
11	0.64953E C1	C.656C9E C1	-C.6565E-C1	-1.006	-0.937
12	0.65035E 01	0.66493E 01	-C.1458E CC	-2.217	-2.082
13	C.662C9E C1	C.66672E C1	-0.4631E-01	-0.697	-0.661
14	0.70831E 01	0.71158E C1	-C.3268E-01	-C.460	-C.467
15	0.73280E 01	0.75019E 01	-0.1740E 00	-2.346	-2.484
16	C.759C6E C1	C.76C34E 01	-C.1280E-01	-0.168	-0.183
17	0.766CCE 01	C.77609E 01	-C.1CC9E CC	-1.309	-1.441
18	C.7C674E C1	0.69520E 01	0.1155E 00	1.648	1.649

ROOT MEAN SQUARE OF PERCENT DIFFERENCES 1.711

STANDARD ERROR OF ESTIMATE 0.123

R = MULTIPLE CORRELATION COEFFICIENT 100(R) = 95.723

Z = 100(DIFF)/(MEAN OBSERVED T1)

MATRIX EQUATION A(B)=C

RECTANGULAR SYSTEMS MATRIX (A/C)

0.18CCCE C2 -C.1F281E C2 0.126C7E C3

-0.18281E 02 0.19260E 02 -0.12668E 02

IDENTITY AUGMENTED WITH SOLUTION VECTOR(S) (I/B)

1.CCCC15 -C.000015 8.982147

0.000015 0.995585 1.947559

Reproduced from best available copy.

INVERTED RSM MATRIX (A-/B)

0.15417E 01 0.14634E 01 0.89821E 01

0.14634E 01 0.14409E C1 0.19479E C1

TRANSFORMED VARIABLES

OBS	T1	T2	T3	T
1	0.784041E 01	-0.695571E 00	0.175205E 01	
2	0.724502E 01	-0.975672E 00	0.210278E 01	
3	0.690223E 01	-0.110721E 01	0.222115E 01	
4	0.694211E 01	-0.999044E 00	0.225512E 01	
5	0.715198E 01	-0.102405E 01	0.219241E 01	
6	0.676155E 01	-0.118564E 01	0.236173E 01	
7	0.688655E 01	-0.120632E 01	0.241044E 01	
8	0.646776E 01	-0.129317E 01	0.241044E 01	
9	0.607868E 01	-0.968828E 00	0.216495E 01	
10	0.673464E 01	-0.110589E 01	0.229623E 01	
11	0.645527E 01	-0.124296E 01	0.242765E 01	
12	0.650352E 01	-0.119758E 01	0.246240E 01	
13	0.662086E 01	-0.118841E 01	0.235276E 01	
14	0.708314E 01	-0.958087E 00	0.213418E 01	
15	0.732797E 01	-0.759871E 00	0.202538E 01	
16	0.759062E 01	-0.707776E 00	0.192788E 01	
17	0.766001E 01	-0.626919E 00	0.186747E 01	
18	0.705744E 01	-0.104221E 01	0.202449E 01	

A VALUES

1)	0.0
2)	0.0
3)	0.0
4)	0.0
5)	0.0
6)	0.0
7)	0.0
8)	0.0

Reproduced from best available copy.

PARTIAL CORRELATION COEFFICIENTS

2	0.957	
3	-0.972	-0.934
	1	2

(Author's note:
i.e. r_{12} = 0.957
r_{13} = - 0.972
r_{23} = - 0.934)

REGRESSION MODEL ----- T1 = B1 + B2*T2 + B3*T3 + ... + BN*TN

B1= 10.46258736

B2= 0.78093910

B3= -1.21920259

Reproduced from best available copy.

OBS	OBSERVED T1	COMPUTED T1	DIFFERENCE ACTUAL	DIFFERENCE %	DIFFERENCE Z
1	0.78494E 01	0.77824E 01	0.6703E-01	0.858	0.857
2	0.72450E 01	0.71390E 01	0.1060E 00	1.474	1.513
3	0.69023E 01	0.68921E 01	0.1022E-01	0.148	0.146
4	0.69421E 01	0.69303E 01	0.1177E-01	0.170	0.168
5	0.71520E 01	0.70042E 01	0.1477E 00	2.087	2.109
6	0.67616E 01	0.66597E 01	0.1019E 00	1.519	1.455
7	0.66865E 01	0.65841E 01	0.1024E 00	1.544	1.462
8	0.64678E 01	0.65163E 01	-0.4854E-01	-0.748	-0.693
9	0.69787E 01	0.70749E 01	-0.9622E-01	-1.369	-1.374
10	0.67346E 01	0.68017E 01	-0.6704E-01	-0.991	-0.957
11	0.64953E 01	0.65345E 01	-0.3928E-01	-0.603	-0.561
12	0.65035E 01	0.65277E 01	-0.2413E-01	-0.370	-0.345
13	0.66209E 01	0.66684E 01	-0.4751E-01	-0.715	-0.678
14	0.70831E 01	0.71145E 01	-0.3138E-01	-0.442	-0.448
15	0.73280E 01	0.73970E 01	-0.6900E-01	-0.937	-0.985
16	0.75906E 01	0.75613E 01	0.2931E-01	0.387	0.419
17	0.76600E 01	0.76980E 01	-0.3804E-01	-0.495	-0.543
18	0.70674E 01	0.71824E 01	-0.1150E 00	-1.614	-1.642

ROOT MEAN SQUARE OF PERCENT DIFFERENCES 1.169

STANDARD ERROR OF ESTIMATE 0.082

R = MULTIPLE CORRELATION COEFFICIENT 100(R) = 98.264

Z = 100(DIFF)/(MEAN OBSERVED T1)

MATRIX EQUATION A(B)=C

RECTANGULAR SYSTEMS MATRIX (A/C)

0.19000E 02	-0.18281E 02	0.39387E 02	0.12607E 03
-0.18281E 02	0.19260E 02	-0.40666E 02	-0.12668E 03
0.39387E 02	-0.40666E 02	0.86917E 02	0.27445E 03

IDENTITY AUGMENTED WITH SOLUTION VECTOR(S) (I/B)

0.999512	0.000488	0.000244	10.462158
0.0	0.999756	-0.000244	0.781250
0.000244	-0.000244	0.999512	-1.218262

Reproduced from best available copy.

INVERTED RSM MATRIX (.A./B)

0.17383E 02	-0.11026E 02	-0.13036E 02	0.10463E 02
-0.11026E 02	0.11288E 02	0.10278E 02	0.78094E 00
-0.13036E 02	0.10278E 02	0.10728E 02	-0.12182E 01

DATA

END

```
M A I D S
DILLINGHAM 1964--BESETT 1966--BRIDGES 1967


C  EXAMPLE NUMBER 2
T  ORDER       =T1          =T1
T              *A2          12.0
T              *A1          *A3
T                           *T2
V*  3
PRINT
L X VALUES
      6.0
      4.0
      1.0
      9.0
      7.0
      3.0
      5.0
      8.0
     10.0
      2.0
A  -2.0        3.0          6.0
*DATA
```

Reproduced from best available copy.

TRANSFORMED VARIABLES

OBS	T1	T2	T3	T
1	0.100000E 01	0.100000E 01	C.700000E 01	
2	C.2C0C0E 01	0.4000C0E 01	0.280000E 02	
3	0.300000E 01	0.7C0C0E 01	C.61CCC0E 02	
4	0.400000E 01	0.100000E 02	C.1C6C00E 03	
5	C.5C0C0E C1	0.130C0E 02	0.163000E 03	
6	0.600000E 01	C.160C0E C2	C.232C00E 03	
7	0.700000E 01	0.190000E 02	0.313000E 03	
8	C.80CCC0F 01	0.2200C0E 02	0.406000E 03	
9	0.9000C0E 01	0.250C00E 02	C.5110C0E 03	
10	0.10C0C0E 02	0.280000E 02	0.628000E 03	

A VALUES

1)	-0.200000E 01
2)	C.3CC000E 01
3)	C.6C0C0E 01
4)	0.0
5)	0.0
6)	0.0
7)	0.0
8)	0.0

DATA

A 5.0 7.0 -3.0

*****

TRANSFORMED VARIABLES

OBS	T1	T2	T3	T
1	0.100000E 01	0.120000E 02	0.500000E 01	
2	0.200000E 01	0.190000E 02	0.700000E 01	
3	0.300000E 01	0.260000E 02	-0.100000E 01	
4	0.400000E 01	0.330000E 02	-0.150000E 02	
5	0.500000E 01	0.400000E 02	-0.350000E 02	
6	0.600000E 01	0.470000E 02	-0.610000E 02	
7	0.700000E 01	0.540000E 02	-0.930000E 02	
8	0.800000E 01	0.610000E 02	-0.131000E 03	
9	0.900000E 01	0.680000E 02	-0.175000E 03	
10	0.100000E 02	0.750000E 02	-0.225000E 03	

A VALUES

1) 0.500000E 01
2) 0.700000E 01
3) -0.300000E 01
4) 0.0
5) 0.0
6) 0.0
7) 0.0
8) 0.0

Reproduced from best available copy.

DATA

STOP

REFERENCES

1. International Business Machines Corporation. Reference Manual C28-6629-0. IBM System/360 Basic FORTRAN IV Language, New York, N.Y. 1963.

2. Dimitry, D. L. and T. H. Mott, Jr. Introduction to FORTRAN IV Programming. Holt, Rinehart, and Winston, Inc., New York, N.Y. 1966.

3. Ostle, Bernard. Statistics in Research. 2nd edition. The Iowa State University Press, Ames, Iowa. 1963.

4. Mills, F. C. Statistical Methods. 3rd edition. Henry Holt and Company, New York, N.Y. 1955.

5. Wylie, C. R., Jr. Advanced Engineering Mathematics. 2nd edition. McGraw-Hill Book Company, Inc., New York, N.Y. 1960.

6. Dillingham, J. R., J. L. Cleasby, and E. R. Baumann, "Prediction of Diatomite Filter Cake Resistance," ASCE, Sanitary Engineering Division Proc. 93: 57-76. 1967.

ACKNOWLEDGEMENT

This work was supported by the Engineering Research Institute, Iowa State University, in part through funds made available by the Federal Water Pollution Control Administration through FWPCA Research Grant WP-00196 and Research Fellowship 5-F1-WP-26,104 (H. R. Bridges) and by the U.S. Army Engineer Research and Development Laboratories through Research Contract DAAK02-67-C-0408.